Monitoring and Modeling Performance of Communications in Computational Grids

Michael A. Frumkin*, Thuy T. Le**
*NASA Advanced Supercomputing (NAS) Division
NASA Ames Research Center, Moffett Field, CA 94035-1000
**San Jose State University
frumkin@nas.nasa.gov, thuytle@email.sjsu.edu

Abstract

Computational grids may include many machines located in a number of sites. For efficient use of the grid we need to have an ability to estimate the time it takes to communicate data between the machines. For dynamic distributed grids it is unrealistic to know exact parameters of the communication hardware and the current communication traffic and we should rely on a model of the network performance to estimate the message delivery time.

Our approach to a construction of such a model is based on observation of the messages delivery time with various message sizes and time scales. We record these observations in a database and use them to build a model of the message delivery time. Our experiments show presence of multiple bands in the logarithm of the message delivery times. These multiple bands represent multiple paths messages travel between the grid machines and are incorporated in our multiband model.

1 Introduction

With computational grids coming into service the quality of assignment of the application tasks to the grid machines, also known as dynamic scheduling [2], or navigation [4], directly affects application turnaround time. The assignment decisions depend on many factors: identification of appropriate grid machines, the load of the machines, the application requirements, the latency and bandwidth of the network, the traffic in the network. To obtain these factors a few tools have been developed, such as `traceroute`, the *Network Weather Service* (NWS) [14], the *NAS Grid Benchmarks* (NGB) [3]. These tools allow one to obtain numerical estimates of the factors affecting the grid scheduling decisions. These estimates represent a sparse subset of all possible loads the grid can accommodate.

We use observations of the message delivery time in a computational grid to build a multiband model of the network which generalizes a single-band model described in [6]. This analytical model can be used for quick and reliable estimation of the time it takes to communicate application data between hosts of a grid. We build the model in two steps. First, we obtain experimental data for the message delivery time between hosts of the grid. Then we extract the band structure of these communications by calculating a histogram of the logarithms of the message delivery times.

As a main measurement tool we use a Java version of the NAS Grid Benchmarks [3, 9]. Using the NGB for measurements has a number of advantages. The Java version is architecture and OS neutral and can be easily used to build a computational grid environment. It does not require users to have accounts on all grid machines, which simplifies our collaborative effort. An assignment of benchmark tasks can be done by simple changes in a benchmark data flow graph that gives great flexibility to concentrate on any interesting subset of the hosts. The benchmarks can be executed in a monitoring mode to build a database of grid measurements, including one-way message transmission time. In the computational grids many different mechanisms for communication between machines may be used, including `MPI`, `Java RMI`, `GridFTP`, `scp` and others. To verify the qualitative results obtained with NGB we compare the results with the measurements obtained with `traceroute` and with `scp`.

2 Observing the Network Traffics

2.1 Traceroute

The `traceroute` utility is the most popular tool for discovery network structure, for finding latencies incurred by messages sent between machines and for diagnostics of network anomalies [10]. For testing a route between hosts

A and B, `traceroute` sends a sequence of test packets (UDP datagram) from A to B until a packet reaches its destination [11, Section 25.6]. The first packet has value of IPv4 *Time To Live* (TTL) field (or IPv6 hop limit field) equal to 1. This packet causes the first router along an (A, B) path to return an Internet Communication Message Protocol (ICMP) "time exceeded in transit" error. The value of the TTL field of the next packet is incremented by 1. Each router along the path of the packet decrements TTL by 1, hence each packet travels one hop farther than the previous one. If a packet reaches B, then the host returns an ICMP "port unreachable" error. These returned error messages allow `traceroute` to find out the IP addresses of the routers where TTL vanishes. The `traceroute` prints these IP addresses and the time elapsed since sending a packet till receiving an error message that the packet have not been delivered.

A number of `traceroute` servers have a web page that allows to find the routers and the latencies along a path from the server to any other internet host. The `traceroute` utility also allows us to find out the bandwidth of an (A, B) path in the range of messages in the interval $[40, 65534]$. Paxson [10] uses extensive experiments with `traceroute` to detect and classify internet anomalies. In other papers they provide evidence that the arrival times of network messages are fractal, i.e. self-similar in a range of time scales [13] and have heavy tails in the distribution of arrival times. One conclusion form this self-similarity is that the arrival times are bursty in an interval of time scales, hence the latency and bandwidth of an (A, B) path (i.e. single band model) have limited value for describing of the arrival times.

2.2 Network Weather Service

Information on TCP/IP performance (latency and bandwidth) can be obtained with the Network Weather Service (NWS). NWS monitors delivery times of messages sent between participating network hosts. The obtained measurements are then used by NWS to estimate the latency and bandwidth and to make predictions of these characteristics. NWS does not correct for the clock skew between hosts. This makes the observation of the one directional message delivery time unreliable and does not allow us to measure network asymmetry.

2.3 Other Tools

A number of tools were developed to extend `traceroute's` ability to display information about the network. These tools, including *3D Traceroute* [12] and *GridMapper* [1], are able to collect statistics about network latency and bandwidth, to obtain geographical information about the hosts and routers, and to visualize these statistics and related network activity. GridMapper can access performance information sources and map domain names to physical locations. It allows one to visualize the layout of the grid and animate activity of the grid hosts and networks.

A direct copy operation of a file across the network by using `ftp` (or `scp` for secure networks) can be used to collect statistics on the time to copy files between grid hosts. If called from the receiving machine, `scp` is blocking (it does not return until the file has been received), hence the execution time of `scp` can be used for measuring time to copy files.

2.4 Stochastic Behavior

The difficulties in understanding network traffic are rooted in the numerous sources of uncertainty and even pathology in the network. There are four main categories of the uncertainty affecting network traffic: topology, metric, events and relative timing. The topological uncertainty affects the (A, B) path taken by different packets of the same message. The path can have fast fluctuation (fluttering), can have loops, and temporary outages (loss of network connectivity) [10]. The metrical uncertainty affects packets delivery time due to varying load on the hosts, routers, and other hardware of the network (switches, exchange points). The events uncertainty affects the sharing of the network elements by different messages. Since many events causing data traffic are external to the network (do not have a causal relation to any network event), a message from A to B will have unknown interference with other messages along its way. And, finally, relative event timing, i.e. a small variation in the timing between different events, can substantially affect the message delivery time. We will consider in this section only two aspects of the metrical uncertainty: network asymmetry and violation of the triangle inequality.

2.5 Network asymmetry

The traffic routing in networks does not guarantee that either the time or the path[1] which a packet travels from A to B are the same as these of a packet that travels from B to A. Moreover, the consecutive packets sent by `traceroute` can follow different routes. Indeed, it is easy to observe that paths (A, B) and (B, A) can be different. For example, at the time of writing (May 2003), the route from `www.slac.stanford.edu` (A) to `www.above.net` (B) had 9 hops, while the route in the opposite direction had 13 hops. Surprisingly, the round-trip times measured from each site are very close to each other. This indicates that a packet sent from A to B is returning back to A along

[1] the sequence of routers

a different path and that the chain of routers traversed on round trips from either server is the same.

The unpredictability of the round-trip times can be observed just by looking at the output of a single `traceroute` command. Quite often the packets with larger TTL are returning faster than the packets with smaller TTL.

2.6 Nonmetrical behavior

Table 1 of `traceroute` measurements between three webservers demonstrates that

$$time(ABOVE.NET, SLAC) + time(SLAC, NAS) < \\ < time(ABOVE.NET, NAS).$$

which is a violation of the triangle inequality. The routing tables are supposed to be built with use of the shortest path tree from each router to each network. This would guarantee that the triangle inequality holds between any three hosts. One source of the violation of the triangle inequality is the use of different "length" functions for building shortest path trees on different routers.

Table 1. Round trip times between `www.above.net`, `www.slac.stanford.edu`, and `www.nas.nasa.gov`. The table entries show the average time while the numbers in parentheses show the maximum deviation (3 measurements per table entry on two consecutive days).

Server Name	SLAC	ABOVE.NET	NAS
SLAC	-	55.9 (0.4)	2.4 (0.1)
ABOVE.NET	55.8 (0.3)	-	68.7 (1.3)
NAS	3.5 (1.2)	68.1 (1.2)	-

3 Using the NAS Grid Benchmarks for Monitoring the Grid

3.1 The NAS Grid Benchmarks

The *NAS Grid Benchmarks* (NGB) [3] were designed to represent typical grid applications, and to test grid functionality, performance of grid services. NGB use a basic set of grid services such as create task and communicate. An NGB instance is specified by a *Data Flow Graph* encapsulating NGB tasks and communications between these tasks. The NGB suite contains four benchmarks: Embarrassingly Distributed (ED), Helical Chain (HC), Visualization Pipeline (VP), and Mixed Bag (MB), see Figure 1. An instance of NGB comprises a collection of slightly modified NPB codes. Each NPB code (BT, SP, LU, MG, or FT) is specified by class (mesh size), number of iterations, source(s) of the input data, and consumer(s) of solution values. The NGB currently specify five classes: S, W, A, B, and C.

3.2 Enhancements to the NGB

For probing and analysis of the message delivery times some enhancements to the NGB implementations released by NAS were necessary. We added monitoring capabilities that allow us to run the benchmarks periodically and save the monitoring results in a database. We also increased the range of NGB message sizes and added a clock synchronization.

3.2.1 Fine control over message sizes

The amount of data the benchmarks tasks send to their successors depends on the class. It varies from 69KB (Class S) to 245MB (class C), giving us a sparse set of message sizes to observe the delivery times. Our first modification was to add some extra data to the array sent by a task to its successors. The message sizes were made controllable by an input parameter to the utility `ngbrun` used to submit the benchmarks. The correctness of the message was checked by each receiver and the benchmarks will not verify if any of the additional messages has incorrect checksum. This flexibility in choosing the message size made it possible to implement grid monitoring with growing message sizes, Section 4.

3.2.2 Grid clock synchronization

Synchronization of clocks of the computers in computational grids is accomplished by means of the (Simplified) Network Time Protocol ((S)NTP). The SNTP allows us to synchronize clocks of the computers in a WAN within an accuracy of a few tens of milliseconds [8]. However, increasing the speed of the network routers and switches allows us to send fast messages between grid machines, so even a clock skew of 10 ms becomes noticeable. Another source of the time skew is improper function/configuration of NTP daemons. In our experiments the clock skew between grid machines manifested itself by apparently negative communication times.

To correct the clock skew we implemented a clock synchronization mechanism which on average reduces the clock skew to less than 20 ms. The clock synchronization task works in asymmetrical networks (networks where the time to send a message from A to B may differ from the time to send the same message from B to A). However, the accuracy of the clock skew correction does not exceed half of the round-trip time of the time stamps. After correcting

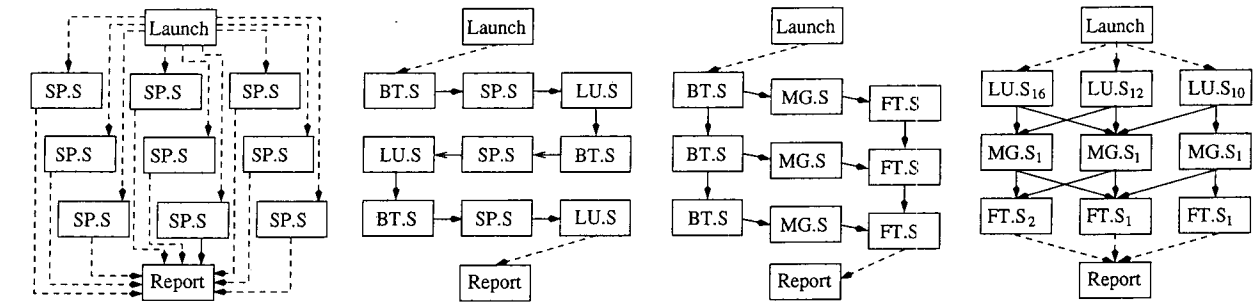

Figure 1. Data flow graphs of NGB, class S. Solid arrows signify data and control flow. Dashed arrows signify control flow only. Subscripts of MB tasks indicate the number of iterations carried out by an NPB code.

the clock skew it is possible to measure asymmetry of the network for the messages whose one way delivery time is significantly bigger than the time stamp round-trip time.

We describe a synchronization of a pair of machines A and B. To synchronize the whole grid we build a spanning tree of the network, then choose a root of the tree and apply the pairwise synchronization to each edge of the tree starting from the root. We use the synchronization mechanism employed by NTP [8] by sending a time stamp $t_A(1)$ from A to B and a time stamp $t_B(2)$ from B to A and record their arrival times $t_B(1)$ and $t_A(2)$ respectively. Let c_{AB} and c_{BA} be the time stamp delivery times (measured by some external clock which we use only for justification of our synchronization mechanism) Let $c = c_{AB} + c_{BA}$ and $C = t_B(1) - t_A(1) + t_A(2) - t_B(2)$ be actual and observed round-trip of a time stamp respectively. Also, let $\delta = c_{AB} - c_{BA}$ and $\Delta = t_B(1) - t_A(1) - (t_A(2) - t_B(2))$ be the actual and observed asymmetry of the delivery time. Let w be the time the clock in B running ahead of clock in A which we assume to be a constant during the period of time c. We have the following relations:

$$t_B(1) - t_A(1) = c_{AB} + w, t_A(2) - t_B(2) = c_{BA} - w.$$

hence, $c = C$ and $2w = \Delta - \delta$. We add $\Delta/4$ to the clock in A and subtract it from the clock in B. If $\delta = 0$ this correction will synchronize the clocks in A and B. In any case the accuracy of the synchronization will exceed $\delta/2 \leq c/2 = C/2$.

4 Experimental Results

For our experiments we used a grid with hosts located in NASA Ames and NASA Glenn research centers, Figure 2. The Java version of the NGB with enhancements was installed on the machines of Table 2. It uses the Java Registry

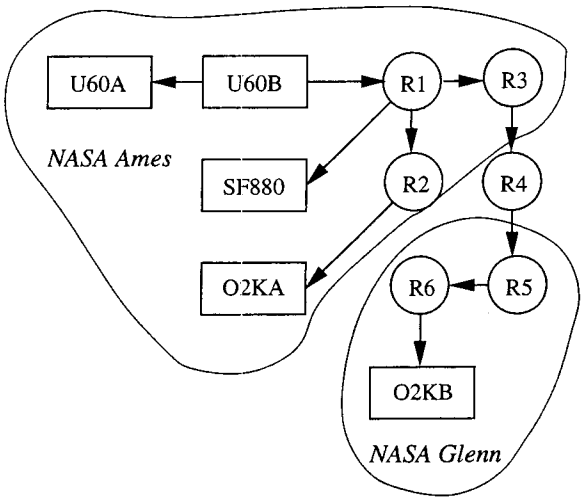

Figure 2. A spanning tree of the experimental grid. The routers R1 through R6 were identified by the `traceroute` ran from U60B . Rectangular blocks show the machines of Table 2.

to register and to lookup task services on the hosts and the Java Remote Method Invocation (RMI) to access the services, run benchmark tasks, and to communicate data between tasks.

We monitored the grid over periods of 24-48 hours for a few weeks. In addition to the NGB we used `traceroute` and `scp`. We used two types of monitoring: with fixed and with growing message sizes, see Table 3. We ranged the monitoring interval between 1 and 30 minutes. Since we have not observed an essential difference over this range, we show results of monitoring with intervals of 10 minutes, unless specified otherwise. The typical time to execute the

Table 2. The the grid machines used in our experiment.

Machine Name	NP	Clock Rate (MHz)	Peak Perf. (GFLOPS)	Memory (GB)	Maker	Architecture	NASA Center
U60A	2	450	1.8	1	SUN	ULTRA60	Ames
U60B	2	450	1.8	1	SUN	ULTRA60	Ames
O2KA	32	250	16	8	SGI	Origin2000	Ames
SF880	8	900	14.4	16	SUN	UltraSparc 3	Ames
O2KB	24	250	12	8	SGI	Origin2000	Glenn

HC.S (i.e. HC, class S) benchmark varies between 20 and 40 seconds in our setup.

Typical monitoring results with fixed message size are shown in Figures 3 and 4. The histograms of the logarithm of the delivery times are shown in the right columns of both figures. All the histograms have multiple extremal points. The *bands* of associated events within 10%-30% probability range surrounding the peaks are shown on the top 2 histograms of Figure 3. To build a model of delivery time of large messages we performed monitoring of the grid with growing message sizes (bandwidth experiments). We started with a series of experiments between U60A and O2KA . The results of experiments using `scp` and HC.S are shown in Figures 5 and 6 respectively. The histograms of the logarithm of the message size over delivery time (i.e. of the logarithm of observed bandwidth) have multiple extremal points and well defined bands (shown in the figures). The results of the bandwidths experiments involving all 4 machines at Ames are shown in Figure 7 while the pairwise results of the experiments involving a machine at Glenn and the machines at Ames are shown in Figure 8. A number of the plots have well separated bands. It is easy to notice asymmetry in the grid, for example, the delivery time O2KA $\Rightarrow$ SF880 is twice as long as the delivery time O2KA $\Leftarrow$ SF880 .

5 Analysis of the Experimental Results

A statistical analysis of the message delivery time obtained with NGB, `traceroute` and `scp` shows that in many cases the histograms of the logarithm of messages delivery times have well separated peaks. This could happen if there are multiple pathes between the hosts and each message travels along one of these paths. If the shortest path is available, the message follows it. Otherwise, if the second shortest path is available the message follows it, and so on. On the other hand, the `traceroute` utility shows that the routes between hosts in our grid remain stable.

To understand the presence of the peaks we consider the arrival time of a word requested by a processor from computer memory. If the word is located in the memory of a MIPS R10000 processor, there are 5 possible cases, as shown in Figure 9 (see [5]). The histogram of the delivery times of words from memory would have 5 peaks. Each peak will have some spread, which is small relative to the separation of the peaks. We need to take the logarithm of the delivery time to take into account a scale of the access times such as on Figure 9.

A delivery of a message from host to host involves a sequence of devices, each having its own scale and bands. Hence, depending on the network traffic a message falls within one of the bands in each device. So the length of the base band is a sum of the base bands of the devices. The second band is a sum of the base bands of all devices except of one with the fastest second band plus the fastest second band. The separation between bands should become smaller as the lower bands get saturated.

6 Conclusions and Future Work

We have performed an extensive set of experiments (over 4000 messages) on a two-site computational grid, using 3 different tools to originate and monitor the messages. The experiments have shown consistency over a time interval of a few weeks. The difference in the message delivery time obtained with `traceroute` and NGB is due to the overhead of interpretation of Java used to implement NGB and to the fact that `traceroute` does not count the delay in the response time of the sending side.

The statistical behavior of the results obtained by all three methods show splitting of the histogram of the logarithm of message delivery time into multiple bands. We explain he presence of the bands with a few different modes in which the messages travel form host-to-host. The base band is observed in the case when there is no interference with network traffic and host processes. As the traffic gets heavier and the host load increases, the base band becomes saturated and the messages are forced through a slower band. We believe that this model of the message delivery explains the presence of heavy tails in the distribution of the message arrival times [7] and answers a question raised in [13, Section 4] regarding the origin of the heavy tails.

We are planning to extend our experiments to larger grids involving some SJSU machines and to use other communi-

Table 3. The message sizes used to monitor message delivery time on the grid.

Monitoring Tool	Fixed Message Size		Growing Message Size		
	size1	size2	initial size	increment	final size
NGB	69KB	-	69KB	400KB	40 MB
`traceroute`	40	64KB	-	-	-
`scp`	-	-	1MB	1MB	40MB

cation methods such as sockets and MPI. We will show how the multiband model can be reduced so we do not have to store $n(n-1)$ pairwise models in a grid having n hosts. Also we will automate the process of building the multiband models.

References

[1] W. Allcock, J. Bester, J. Bresnahan, I. Foster, J. Gawor, J.A. Insley, J.M. Link, M.E. Papka. *GridMapper: A Tool for Visualizing the Behavior of Large-Scale Distributed Systems.* Proceedings of HPDC11, 23-26 July 2002, Edinburgh, Scotland, pp. 179-187.

[2] F. Berman. "High-Performance Schedulers." In: *The Grid. Blueprint for a New Computing Infrastructure.* I. Foster, C. Kesselman, Eds., Morgan Kaufmann Publishers Inc., San Francisco, CA, 1999.

[3] M. Frumkin, Rob F. Van der Wijngaart. *NAS Grid Benchmarks: A Tool for Grid Space Exploration.* Cluster Computing, Vol. 5, pp. 247-255, 2002.

[4] M. Frumkin, R. Hood. "Navigation in Grid Space with the NAS Grid Benchmarks" *Proceedings of the 14th IASTED International Conference "Parallel and Distributed Computing and Systems" (PDCS'2002)*, Cambridge, USA, Nov 4-6, 2002, pp. 24-31.

[5] M. Frumkin, H. Jin, J. Yan. *Automation of Data Traffic Control on DSM Architectures.* Proceedings of International Conference on Computational Science, May 28-31, 2001, San Francisco, CA, LNCS 2074, p. 771-780.

[6] Thuy T. Le. "Evaluating Communication Performance Measurement Methods for Distributed Systems" *Proceedings of the 14th IASTED International Conference "Parallel and Distributed Computing and Systems" (PDCS'2002)*, Cambridge, USA, Nov 4-6, 2002, pp. 45-51.

[7] W.E. Leland, M.S. Taqqu, W. Willinger, D.V. Wilson. *On the Self-Similar Nature of Ethernet Traffic.* IEEE/ACM Transactions on Networking 2(1994), pp. 1-15.

[8] D.L. Mills. *Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI.* RFC 2030 (http://rfc.net), 18 pp., 1996.

[9] The NAS Grid Benchmarks. See http://www.nas.nasa.gov/Research.

[10] V. Paxson. *End-to-End Routing Behavior in the Internet.* IEEE/ACM Transactions on Networking 5(5), pp. 601-615, 1998.

[11] W.R. Stevens *UNIX Network Programming.* Vol. 1, Prentice-Hall International, Inc. 1998.

[12] *3D Traceroute.* www.hlembke.de/prod/3dtraceroute/. 2002-12-29.

[13] W. Willinger, V. Paxson, M.S. Taqqu. *Self-Similarity and Heavy Tails: Structural Modeling of Network Traffic.* A Practical Guide to Heavy Tails: Statistical Techniques and Applications. R. Alder, R. Feldman, M. S. Taqqu, Ed. Birkhauser, Boston 1998.

[14] R. Wolski, N.T. Spring, J. Hayes. "The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing." UCSD Technical Report Number TR-CS98-599, 1998, http://nws.npaci.edu/NWS/.

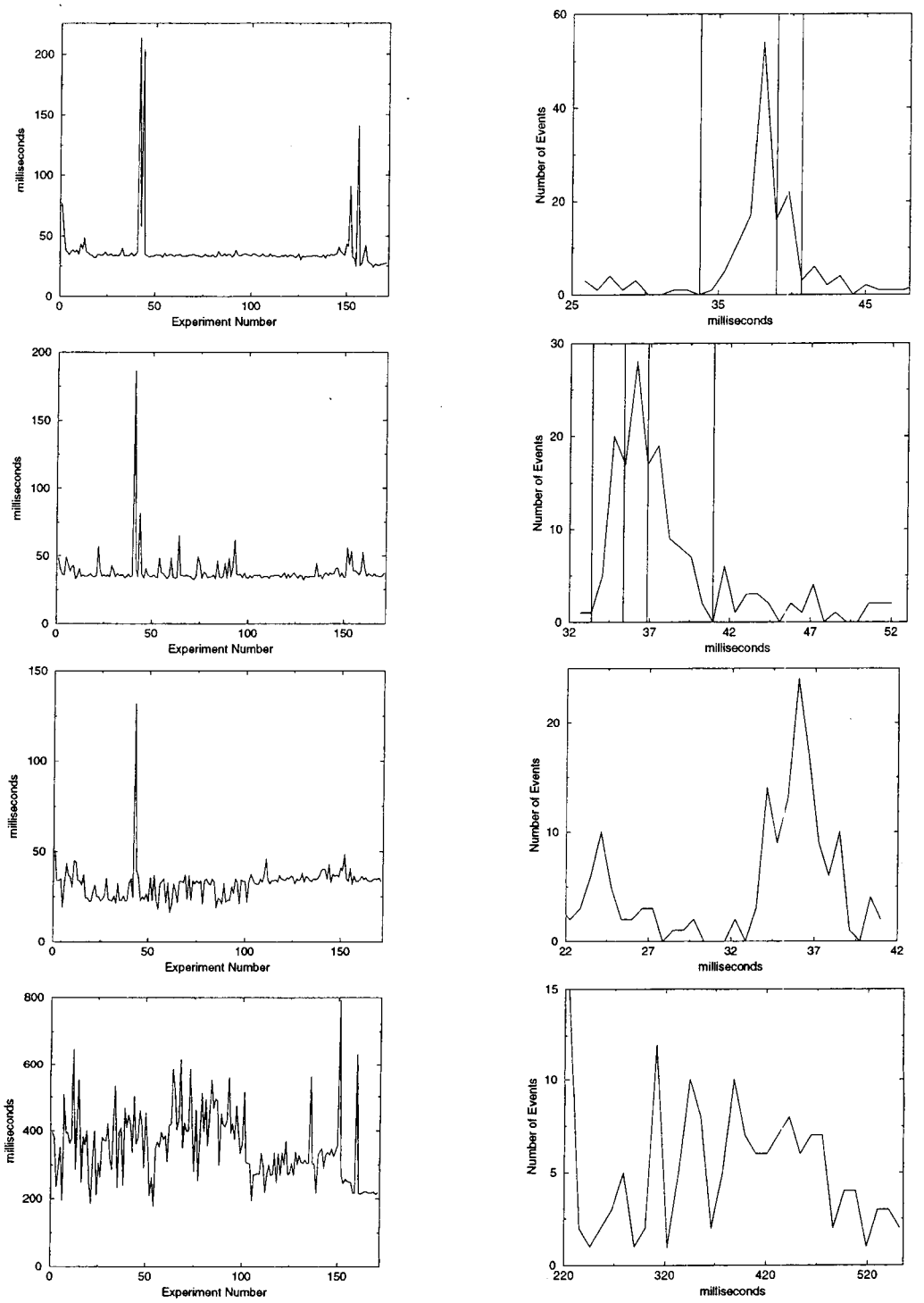

Figure 3. The graphs in the left column from top to bottom show messages delivery times U60A ⇒ O2KA , U60B ⇒ O2KA , O2KA ⇒ SF880 , SF880 ⇒ O2KB respectively. The measurements were obtained by running HC.S of the NAS Grid Benchmarks with 69KB messages. The histograms in the right column show the bands in the distribution of the logarithm of message delivery times (mapped back to linear time scale).

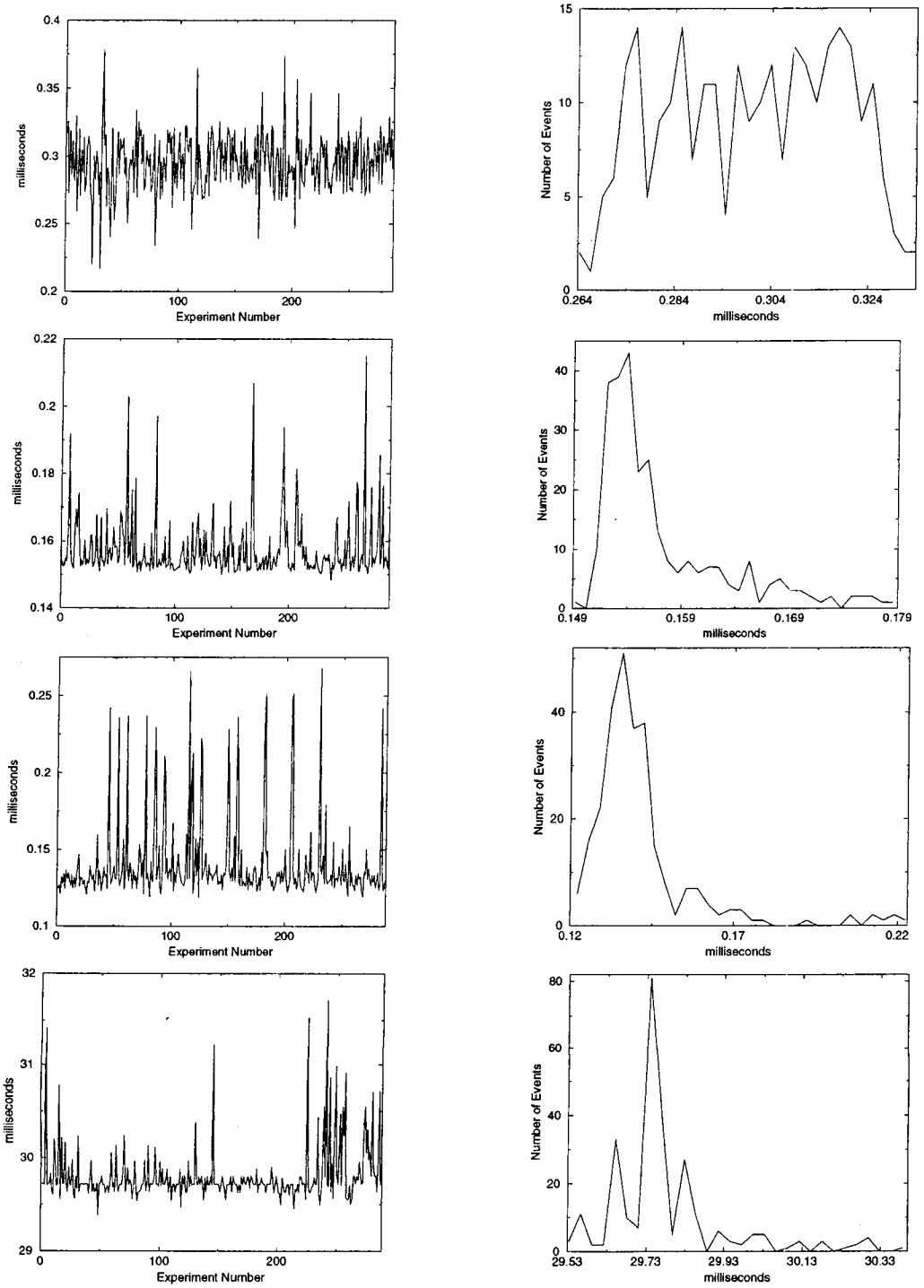

Figure 4. Half of round-trip time of 40 B messages between machines U60A ⇔ O2KA , U60A ⇔ SF880 , U60A ⇔ U60B , U60A ⇔ O2KB respectively. The measurements were obtained by `traceroute`. The histograms on the right expose the bands in the distribution of the logarithm of message round-trip times (mapped back to linear time scale).

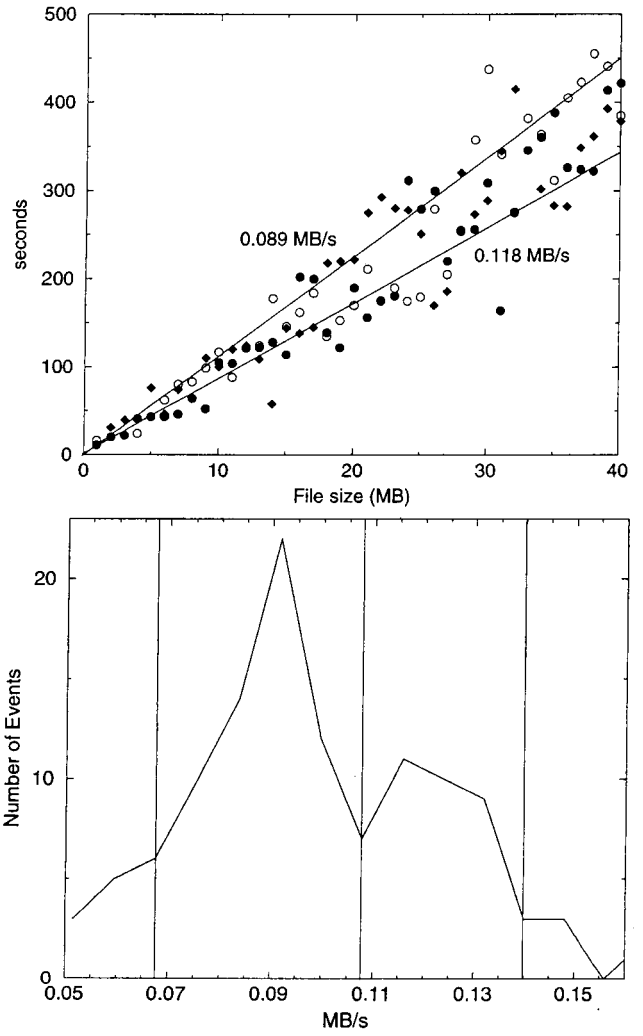

Figure 5. Three sets of experiments on time to copy files U60A ⇒ O2KA using `scp`. The file sizes were in the range [1,40] MB with 1MB increments. The histogram shows 2 bands of 0.091 MB/s and 0.116 MB/s.

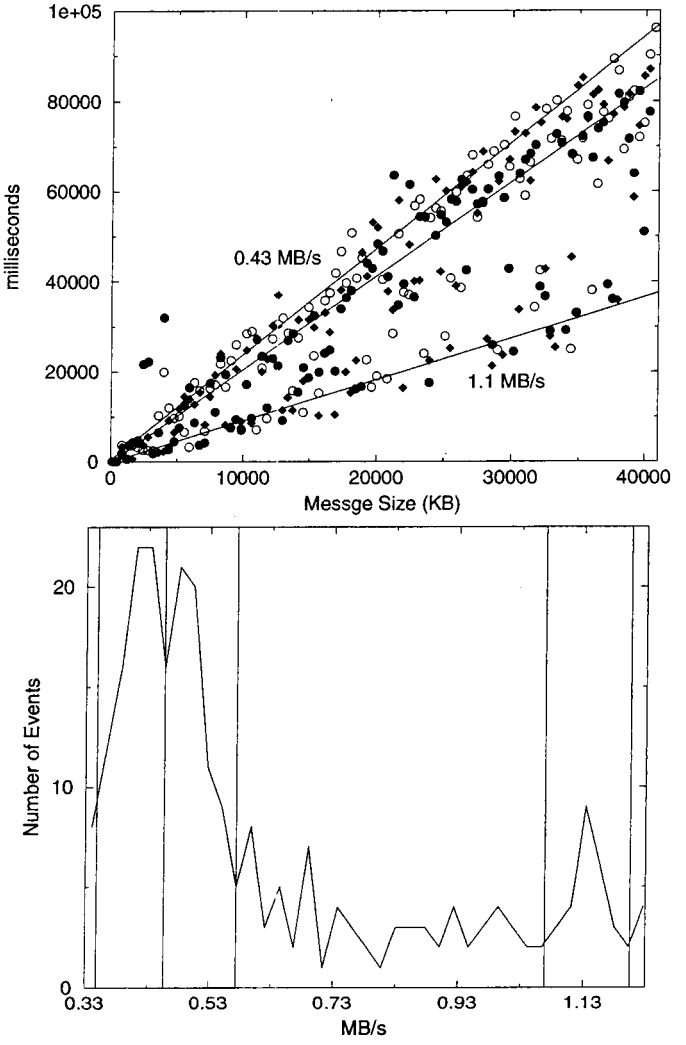

Figure 6. Three sets of experiments on message delivery time U60A ⇒ O2KA . The message sizes were in the range [0.069,40] MB with 0.4MB increments. The histogram shows 3 bands of 0.43 MB/s. 0.49 MB/s, and 1.1 MB/s.

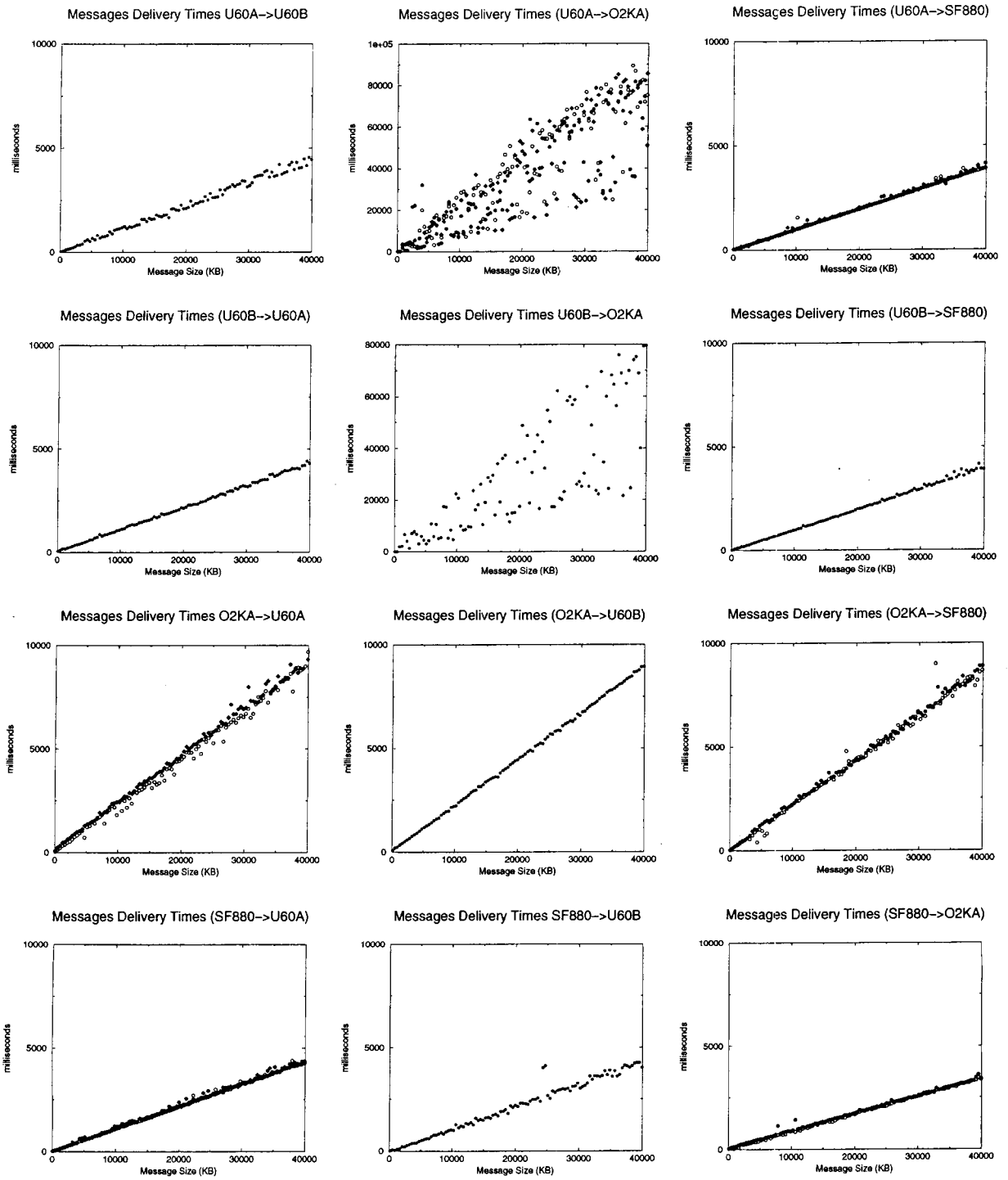

Figure 7. Messages delivery times between U60A , U60B , O2KA , SF880 obtained with the HC.S and the messages varying 69KB-40MB in size (note 10x difference in Y-scale on the plot of times from U60A , U60B to O2KA).

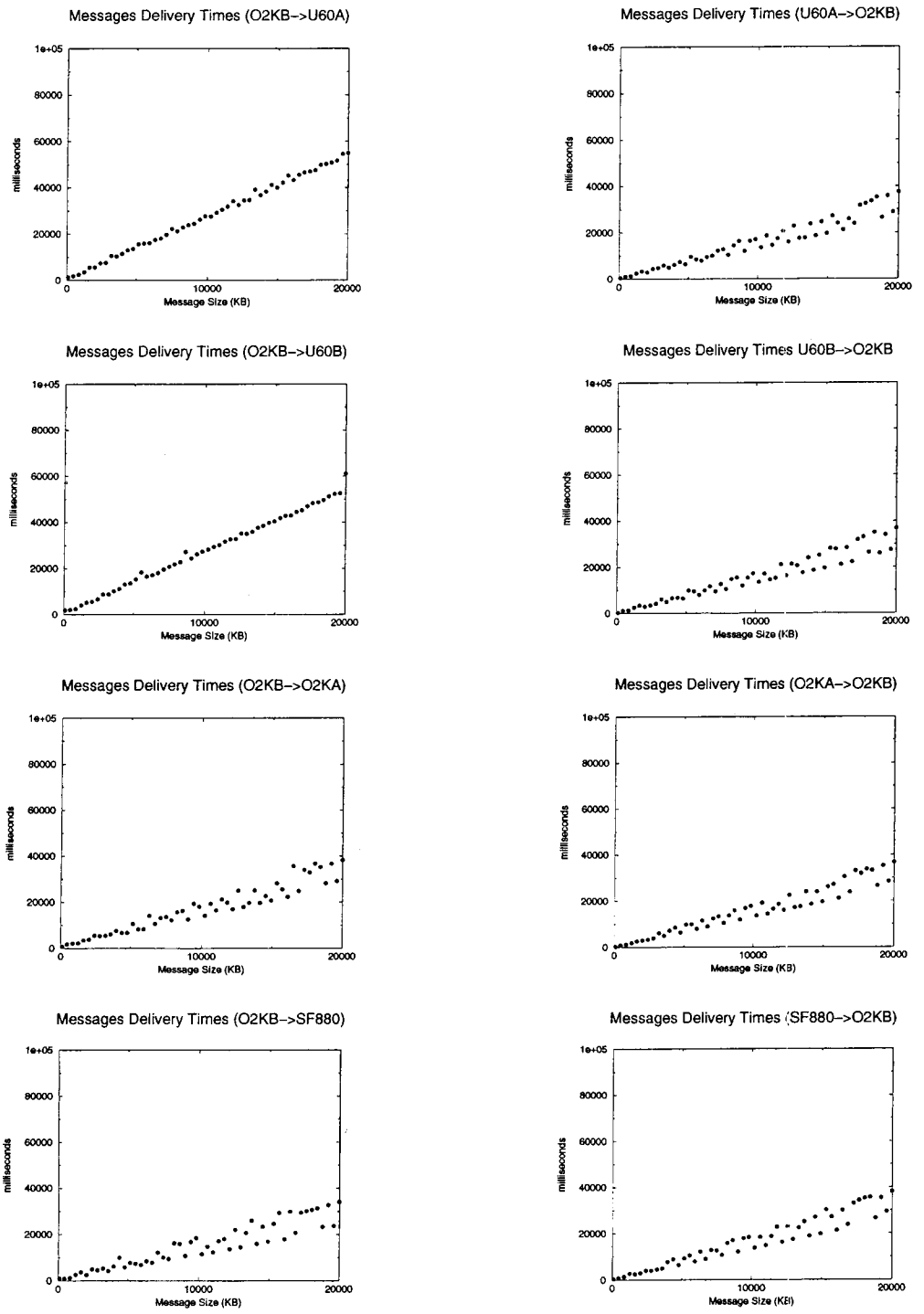

Figure 8. Messages delivery times from/to O2KB to/from other machines (U60A , U60B , O2KA , SF880) obtained with the HC.S and the messages varying 69KB-20MB in size.

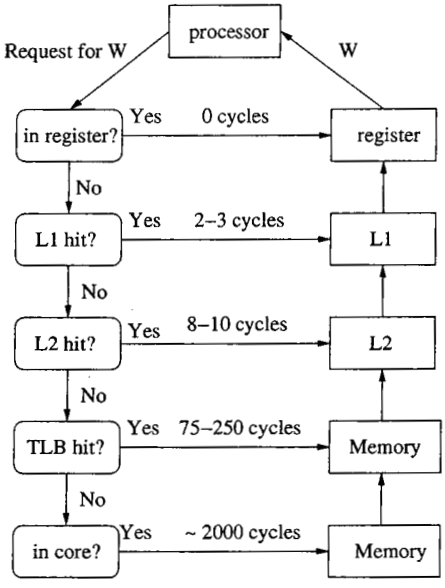

Figure 9. Access time to a word in memory of a MIPS R10000 processor has five bands.